

TCMT software status

Guilherme Lima



NORTHERN ILLINOIS
UNIVERSITY

Calice Workshop at Argonne
March 17-20, 2008

Tcmt software

- Needed functionality similar to the AHcal (then mostly implemented)
 - Good: save development time, reuse some resources and get most future Ahcal enhancements “for free” (e.g. integrated calibration)
 - Bad: potentially affected by any Hcal changes (e.g. HcalTileIndex), and unnatural use of some Hcal-specific implementation details
- Released into the official calice packages in August 2007
 - One round of official reco processing (v0406)

Tcmt vs. Hcal differences

- MappingAndAlignment: chip,channel <--> x,y position of cell in module and layer/module relationships are basic for Hcal and not true for Tcmt
 --> **Ecal/Hcal: ModuleConnection** and **Tcmt: TcmtConnection**
- Calibration constants are keyed by cellkey
 - Hcal: cellkey=(module#chip#channel), where module=1..N
 - Tcmt: module=0 and cellkey = (cassette#strip)
- Tcmt-specific behavior is triggered by module==0:

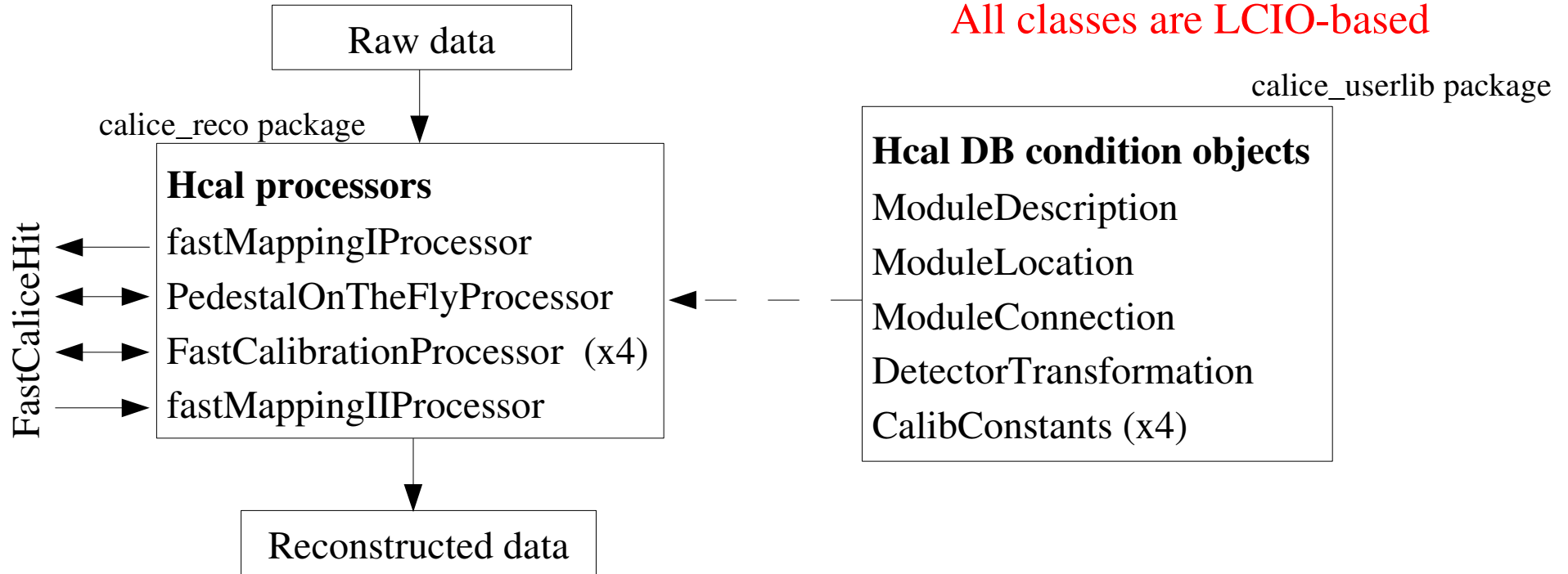
```
void FastCaliceHit::print(std::ostream& os) {
    if(getModule()!=0) {
        // HCAL case
        os << " module: " << std::hex << getModuleID() << std::dec
            << ", chip: " << getChip() << ", channel: " << getChannel();
    }
    else {
        // TCMT case
        os << " cassette=" << getChip() << " strip=" << getChannel();
    }

    os << ", E=" << getEnergyValue() << "+-" << getEnergyError()
        << ", time stamp: " << getTimeStamp() << std::endl;
}
}
```

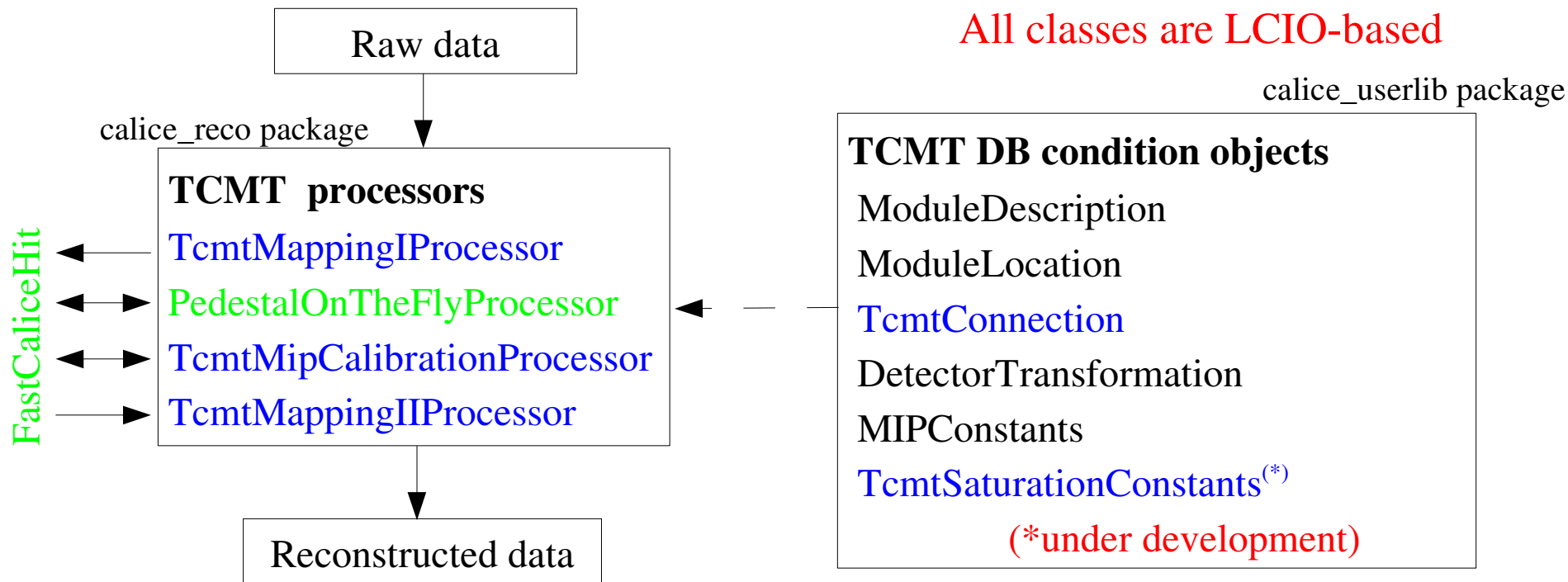


Hcal software structure

All classes are LCIO-based



TCMT implementation



Black: reused from Hcal without changes

Green: reused from Hcal with minimal changes

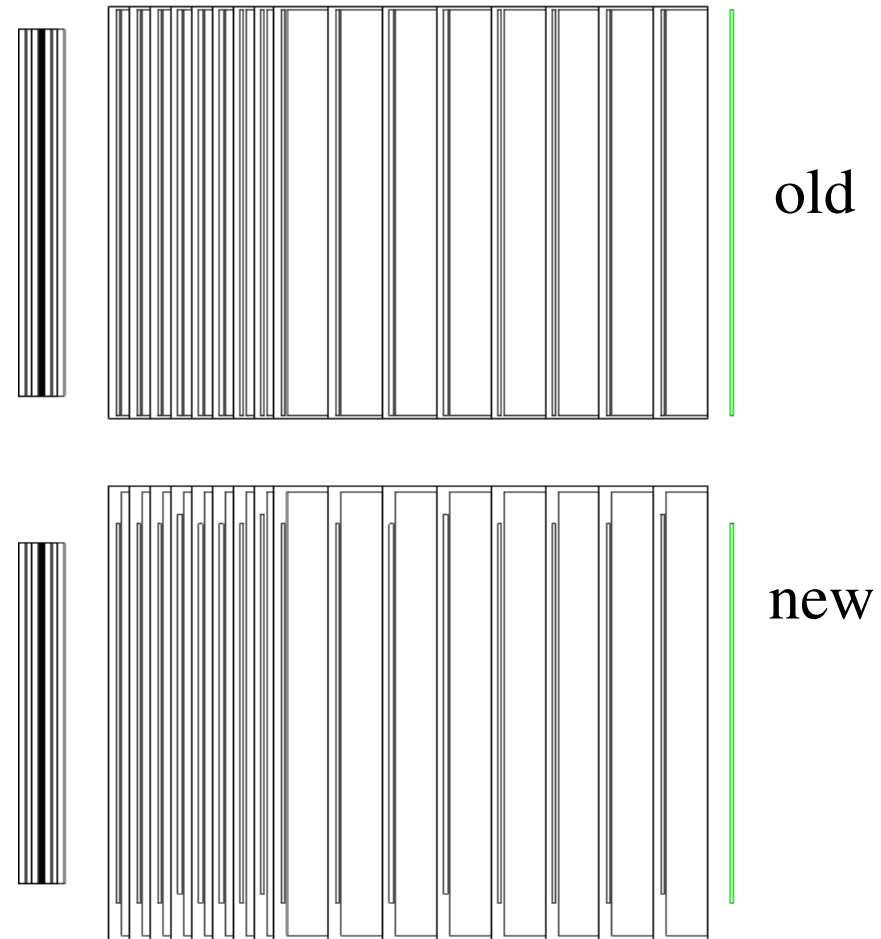
Blue: new code for TCMT, but similar to Hcal

TCMT implementation (2)

- Tcmt conditions DB folders:
 - /cd_calice_cernbeam/Tcmt/TcmtDescription (class ModuleDescription)
 - /cd_calice_cernbeam/Tcmt/TcmtLocation (class ModuleLocation)
 - /cd_calice_cernbeam/Tcmt/TcmtConnection (class TcmtConnection)
 - /cd_calice_cernbeam/Tcmt/TcmtDetectorPosition (class DetectorTransformation)
 - /cd_calice_cernbeam/Tcmt/Mip00 (class MIPConstants)
 - /cd_calice_cernbeam/Tcmt/Saturation00 (class TcmtSaturationConstants) (*not yet, but soon!)
- Binaries to create these DB folders are released into the cddata package:
 - createTcmtDescription, createTcmtLocation, createDetectorTransformationSimple, createTcmtConnection, createTcmtMIPCalibration, createTcmtSaturationConstants

Mokka: TCMT model for TBCern07

- New Mokka driver TBcatcher06 is ready for simulations of CERN July'07 running period
 - Staggering of TCMT modules (hardcoded!):
 - For horizontal strips:
 - layers 2,6,10,14: nominal (y_{nom})
 - layers 4,8,12,16: $y' = y_{nom} + 2.54\text{cm}$
 - for vertical strips:
 - layers 1,5,9,13: nominal (x_{nom})
 - layers 3,7,11,15: $x' = x_{nom} + 2.54\text{cm}$
 - Absorbers have been enlarged
 - Proper positioning after global rotations



TCMT side views

The digitization process

- MC comparisons to data is split into several separate steps:
 - **AHcal:** [noise extraction (data)] + Mokka + ganging + “integrated digitization” (uncalibration, crosstalk, readout smearing, raw noise overlay) + “integrated calibration” + hit selection
 - **TCMT:** [noise extraction + mip calibration (data)] + Mokka + DigiSim (ganging, crosstalk, readout smearing, hit selection + McMip calibration) + noise overlay + hit selection
- Final hit selection: $E_{hit} > 0.5\text{mip}$

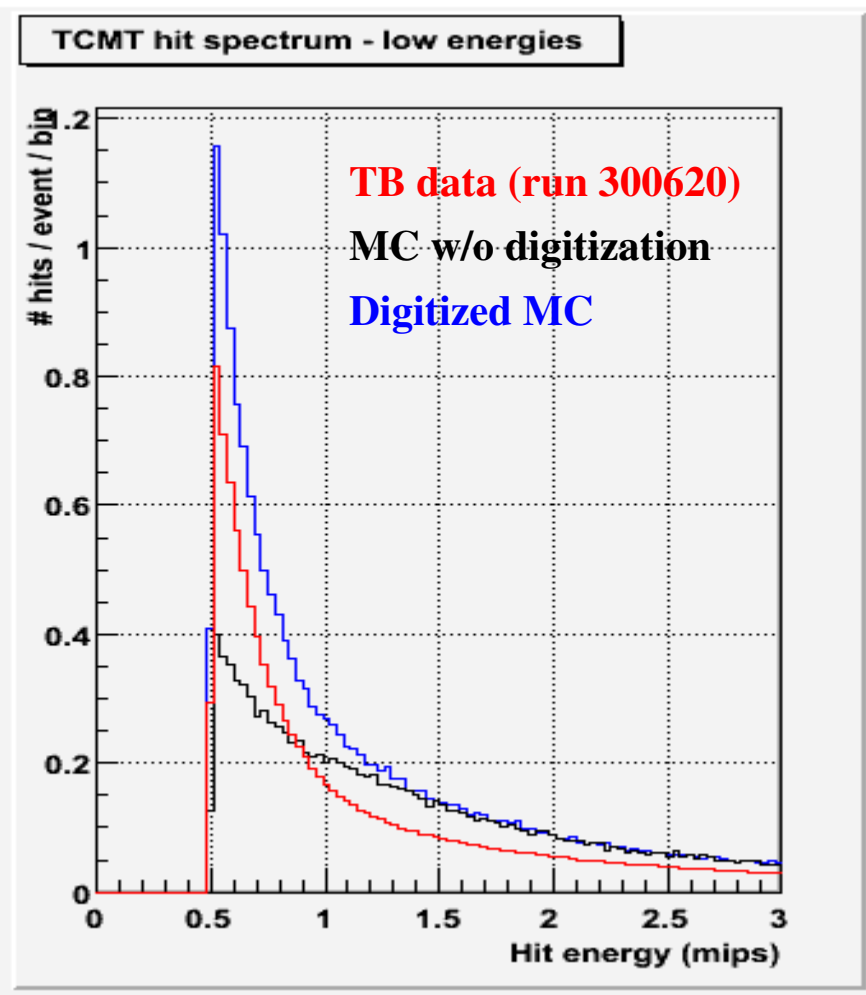
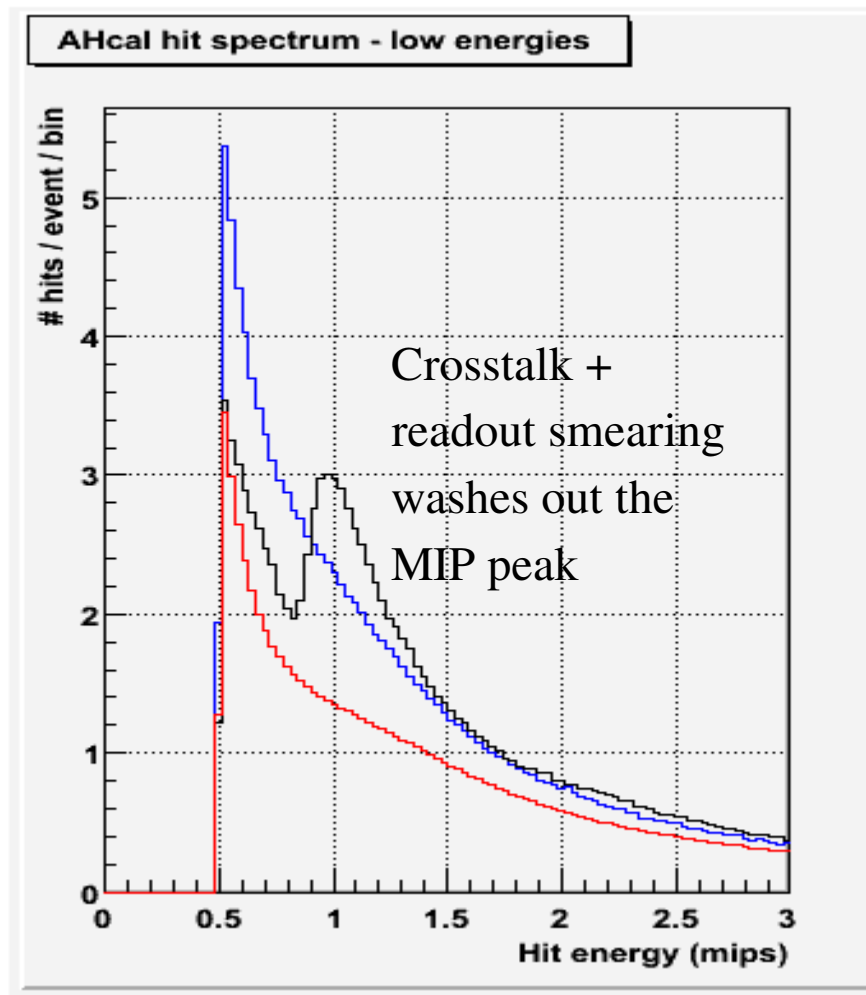
The AHcal digitization simulation procedure

- Noise is combined with MC hits at ADC-counts level, to properly account for saturation corrections
- The procedure is split into following steps:
 - Noise events filtered from TB data (“level2” hits = pedestal-subtracted ADC)
 - MC-ganging processor (simulated virtual cells into real tiles)
 - OverlayProcessor: append noise collections from TB data into the MC event
 - IntegratedHcalDigitizationProcessor: merge hits into a single collection
 - MC hits: physics to electronics mapping, GeV to mips, crosstalk, smeared, *decalibrated* (converted to ADC counts) and combined with noise
 - IntegratedCalibrationProcessor: recalibrate combined hits into mip scale

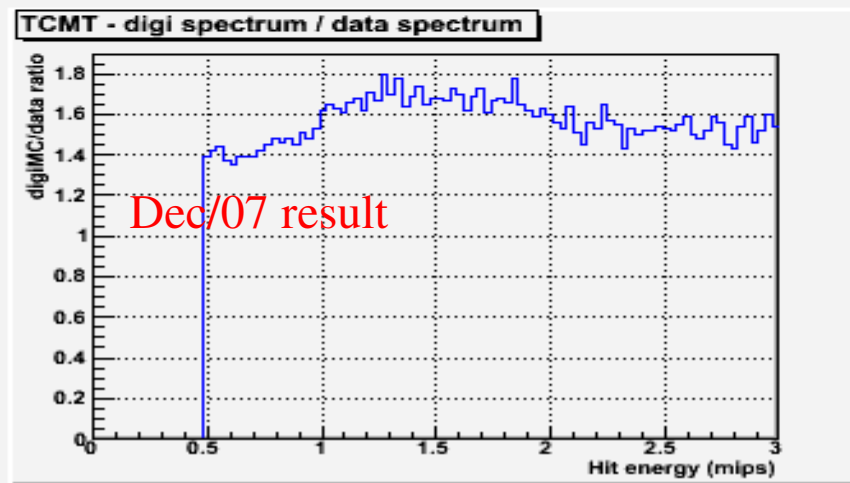
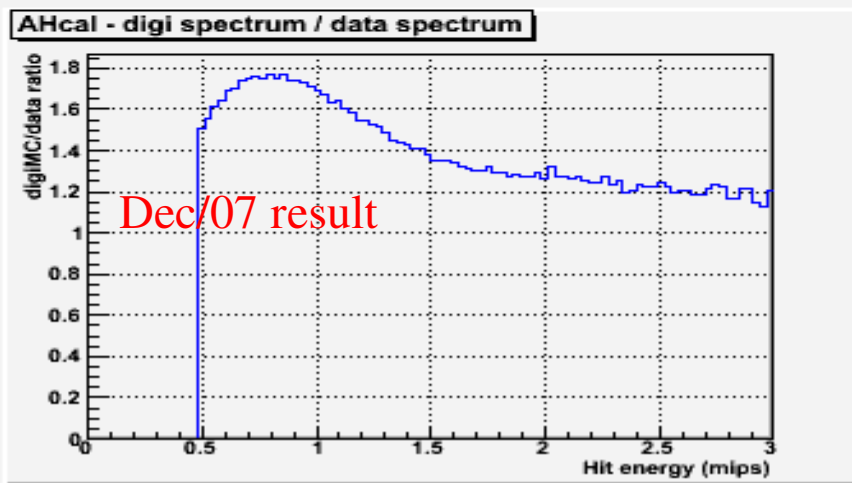
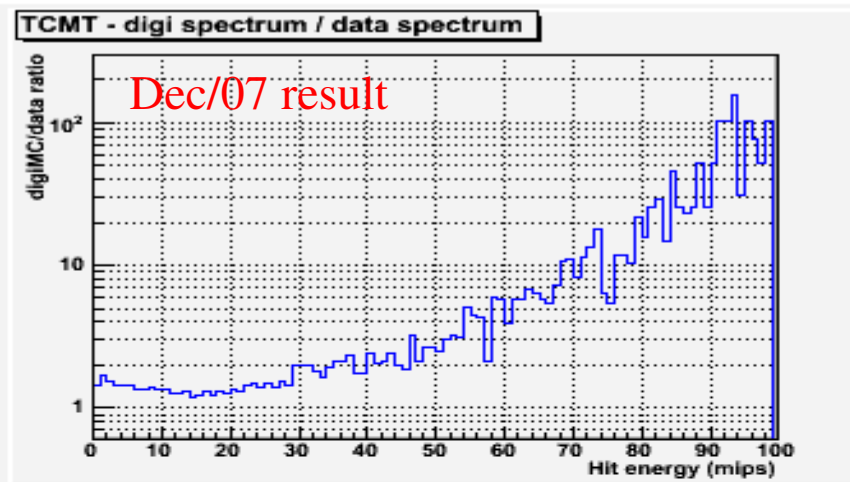
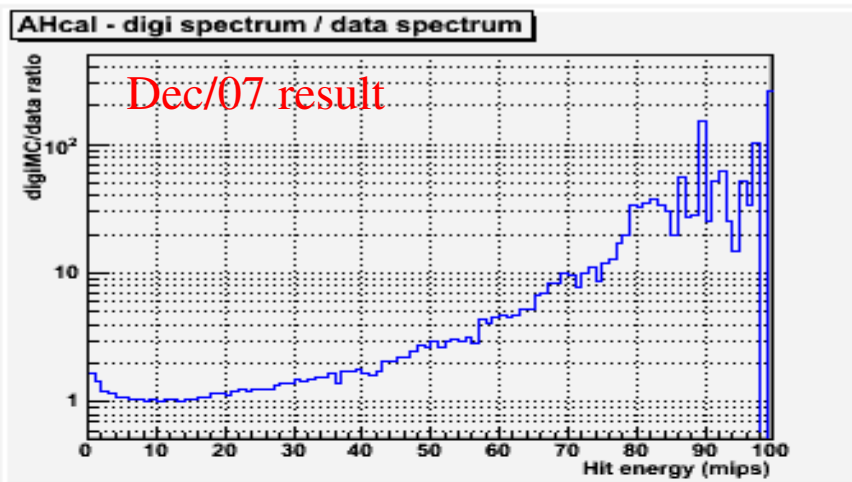
The TCMT digitization simulation procedure

- Recent developments (Oct-Dec/2007)
 - Noise: pedestal triggers filtered from TB data and mip-calibrated
 - DigiSim on MC data: ganging processor, optical crosstalk, smearing, simple mip calibration (no saturation simulation)
 - OverlayProcessor: copy noise collections from TB data into the MC event
 - TcmtOverlayProcessor: combine mip-calibrated noise + mip-calibrated MC hits

Hit spectrum @ low E: 20 GeV pi-

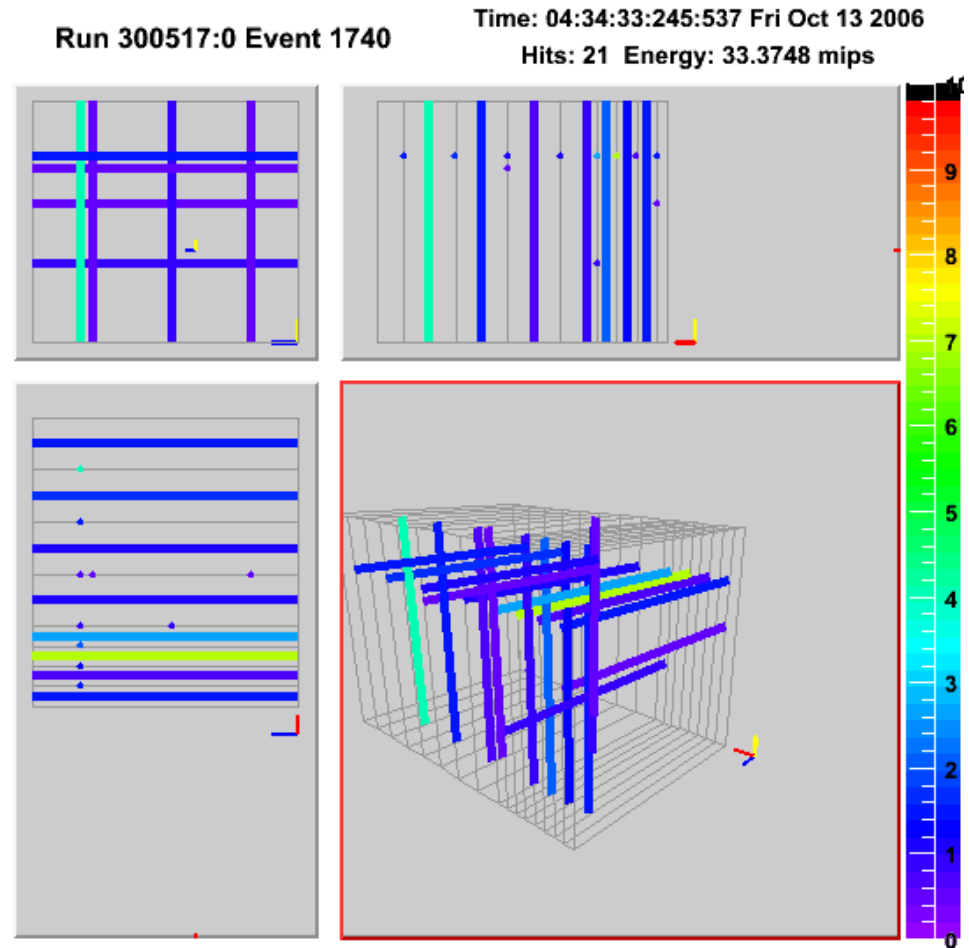


DigiMC/data ratio: 20 GeV pi-



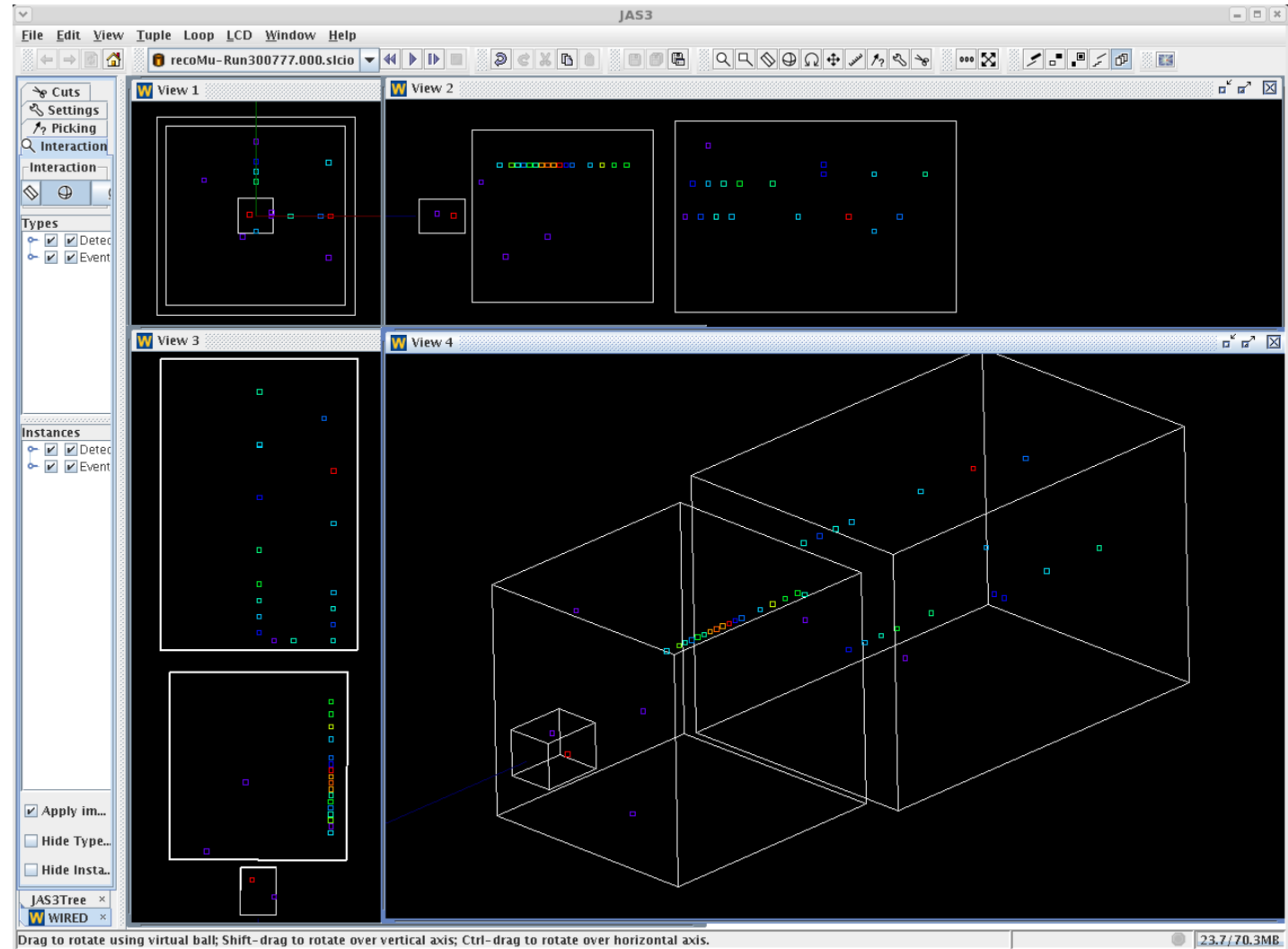
Tcmt in the online monitor

- First significant part of TCMT software to be operational (Aug/06)
- Used for Tcmt commissioning, calibration and early analyses
- No significant updates since then (staggering, rotations)



JAS3 + wired for event display

- Convenient and powerful system
- For full calice support (DB access, rotations, etc), some more development work required, with Java expertise.



Summary

- Most of the Tcmt code has been released into the official CVS repository
Exception is the code for saturation correction of the Tcmt hits, which is under tests.
- Coarse alignment checks done. Are any refined alignments needed?
 - Oct'06: $\sim O(\text{few cm in } x,y \text{ and } \sim \text{mm in } z)$
 - July'07: $\sim O(\text{few cm in } x,y \text{ and } \sim \text{cm in } z)$
- Conditions database ready for reprocessing Cern data (Aug06, Oct06, Jul07)
- Some documentation under preparation